

Microcontroller (Arduino)

MUHAMMAD A. SHEHAB

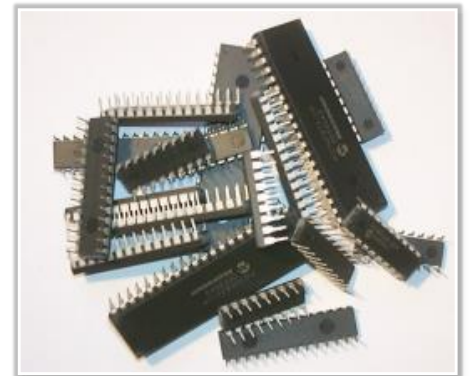
Microcontroller

Why !!



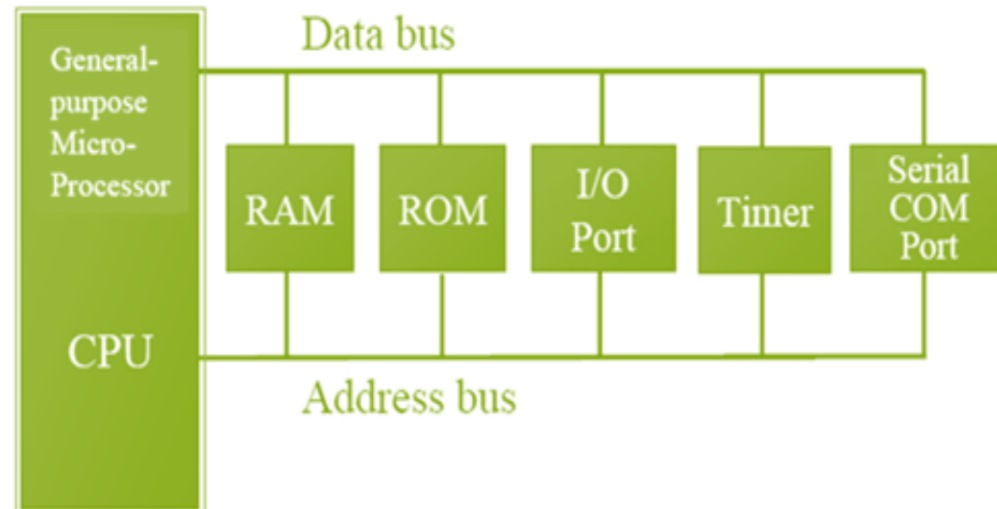
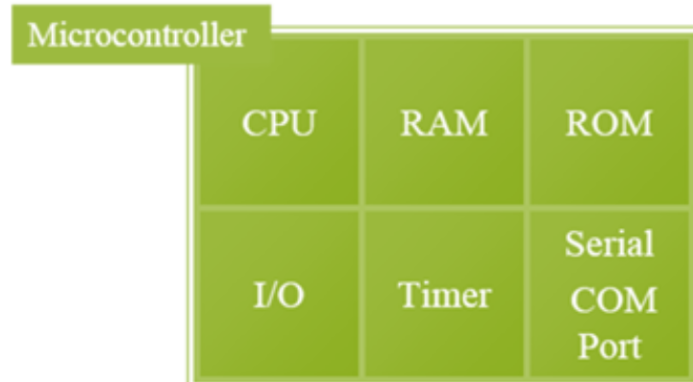
Microcontroller

- ▶ It is a micro-computer. As any computer it has internal CPU, RAM, IOs interface.
- ▶ It is used for control purposes, and for data analysis.
- ▶ Famous microcontroller manufacturers are Micro-Chip, Atmel, Intel, Analog devices, and more.



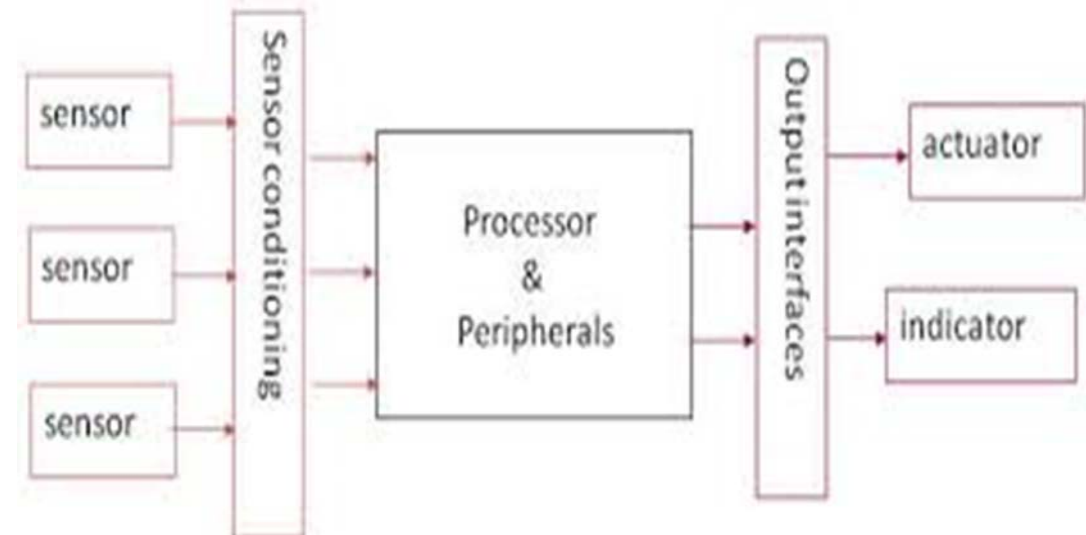
Microcontroller vs. Microprocessor

- ▶ Microcontroller is a small computer on a single chip designed for embedded applications in contrast to microprocessor of general purpose computer
- ▶ Microcontroller vs. Microprocessor



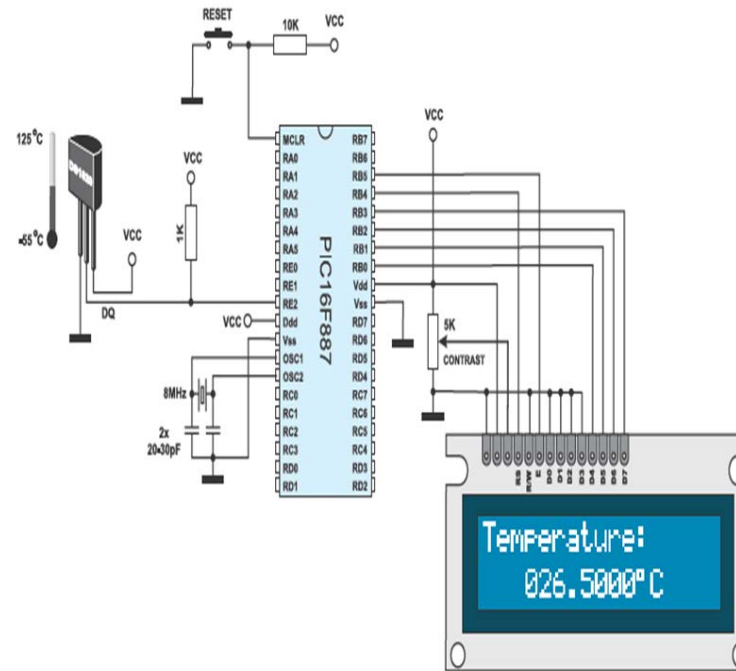
Embedded system

- ▶ A computer system with a dedicated function within a larger mechanical or electrical system. Embedded system is always designed to control the operation and performance of the original system.
- ▶ **Microcontroller** is the **CORE** of any **Embedded system**.



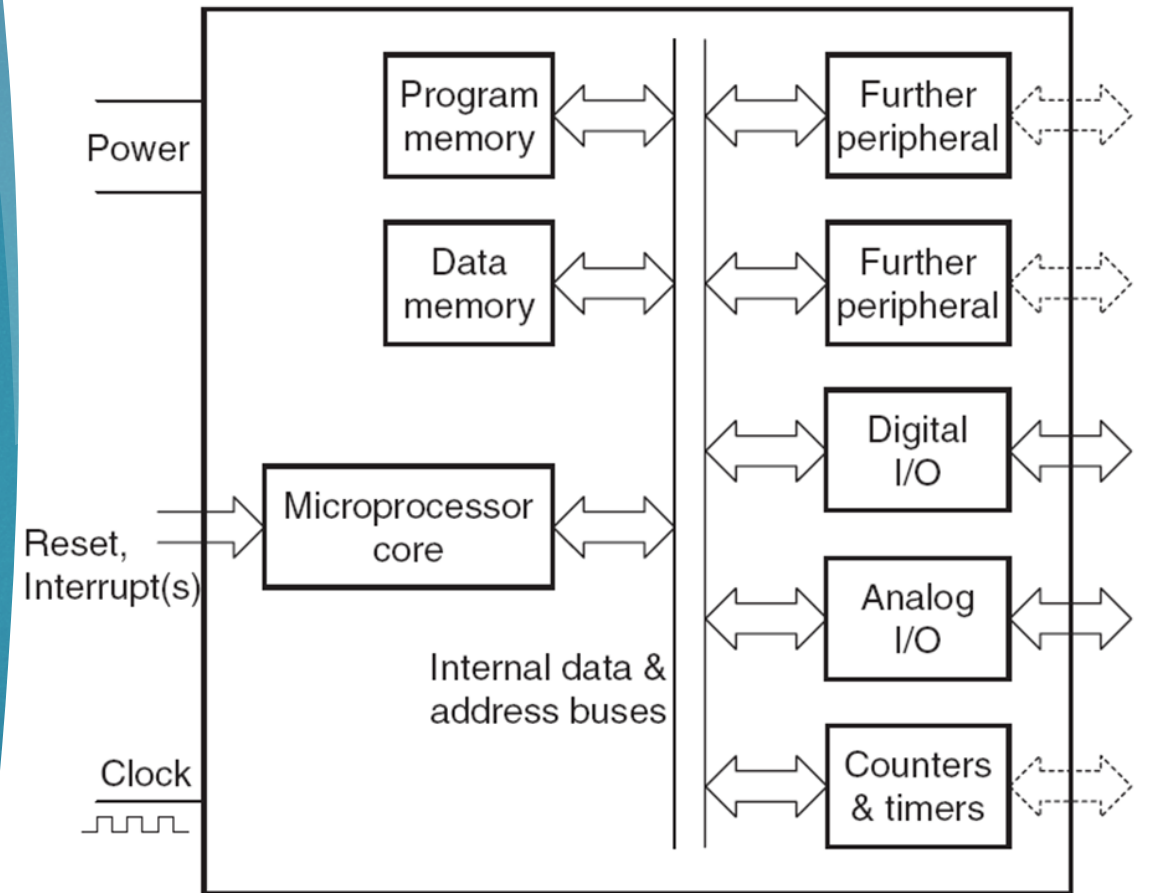
Microcontroller

How !!



- ▶ Microcontroller in Real world
 - ▶ Microcontroller Architecture
 - ▶ Application
 - ▶ Microcontroller choice
 - ▶ Programming
 - ▶ Input/output Interface
 - ▶ Electric & Electronic circuits
 - ▶ Signal Conditioning

Microcontroller Architecture



Microcontroller Choice

- ▶ Speed
- ▶ Packaging
- ▶ Power consumption
- ▶ The amount of RAM and ROM on chip
- ▶ The number of I/O pins and the timer on chip
- ▶ How easy to upgrade higher-performance or lower power-consumption versions
- ▶ Cost per unit

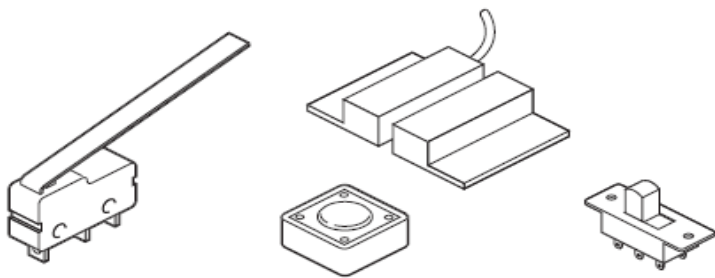
Microcontroller application program

- ▶ Machine Language
 - ▶ Binary Instructions
 - ▶ Difficult to write, find errors, modifications
 - ▶ All programs converted into machine language for execution
- ▶ Assembly Language
 - ▶ Machine instructions represented in mnemonics
 - ▶ Efficient execution and use of memory
 - ▶ Machine-specific
- ▶ High-Level Languages
 - ▶ BASIC, C, and C++
 - ▶ Machine independent
 - ▶ Larger memory and less efficient execution
 - ▶ Easy to write and troubleshoot

Microcontroller input / Output Interface

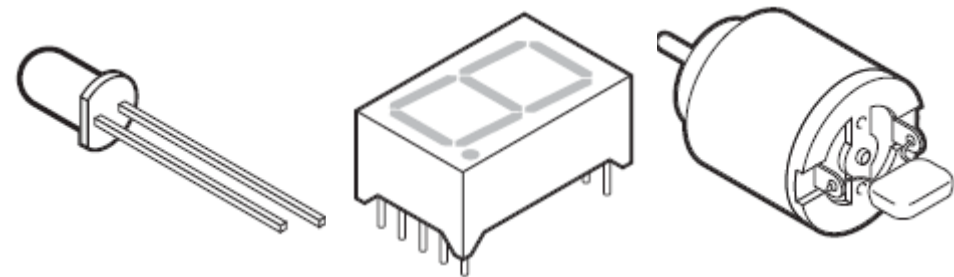
Input Devices

- ▶ Switches
- ▶ Sensors



Output devices

- ▶ Indicators
- ▶ Actuators



How to Burn Code on chip



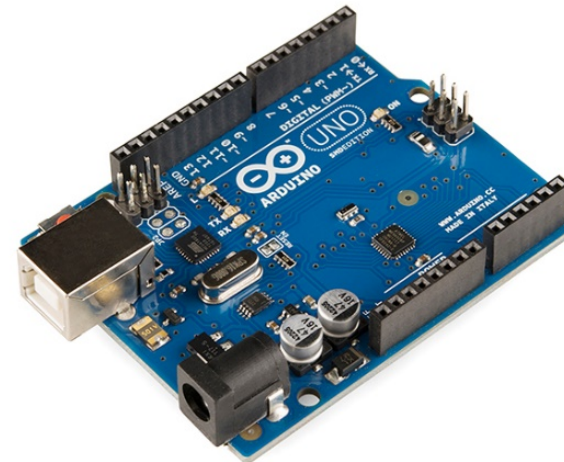
Arduino



Open Source Platform

Arduino

- ▶ A **microcontroller** board, contains on-board power supply, USB port to communicate with PC, and an Atmel microcontroller chip.
- ▶ Used for Rapid prototyping



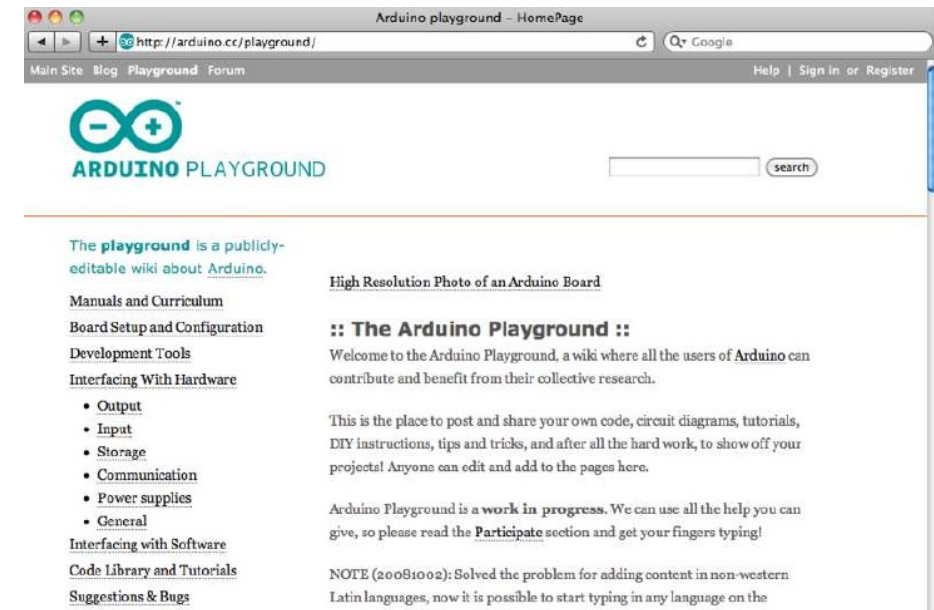
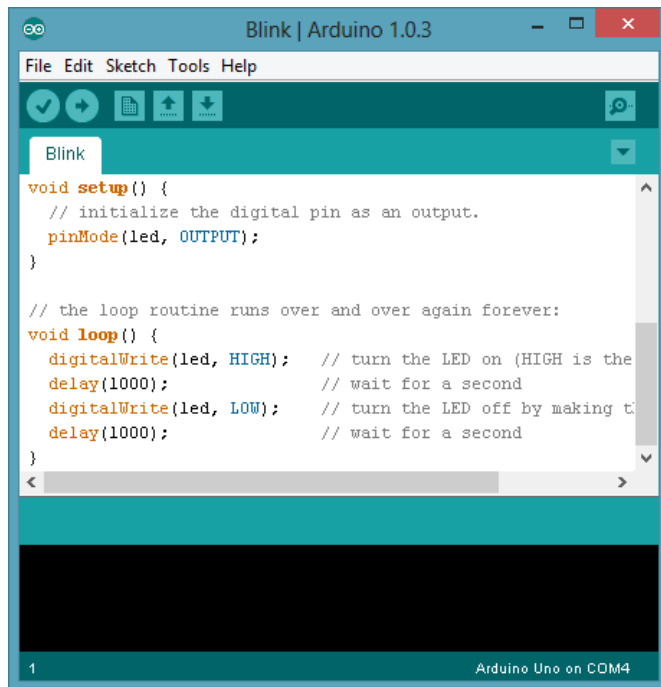
Arduino

Why !!

- ▶ simple and accessible user experience
- ▶ Inexpensive
- ▶ Cross-platform
- ▶ Open source and extensible hardware
- ▶ Open source and extensible software (expanded through C libraries).
- ▶ Simple, clear programming environment (Arduino Software (IDE)).

Arduino

- Integrated SW Libs + Standardized HW + Huge community



Community / The Arduino Playground

Related Hardware and

Playground Content Tree

Arduino

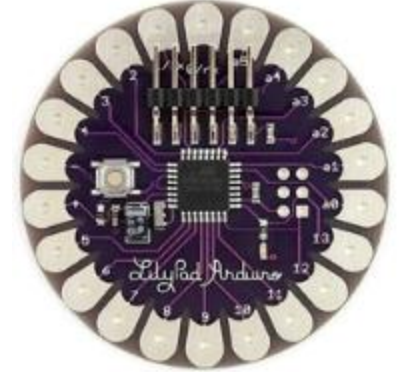
An open-source prototyping platform based on easy-to-use hardware and software.



UNO



Mega



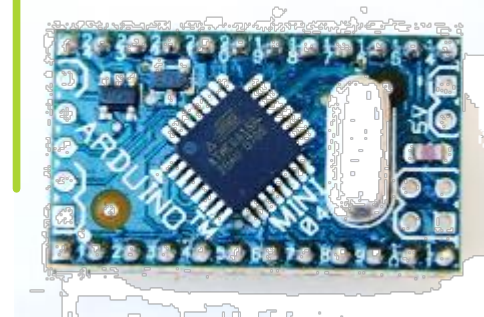
LilyPad



Arduino BT



Arduino Nano



Arduino Mini

Arduino UNO:

Digital output
~: PWM.
0,1: Serial port.

USB port

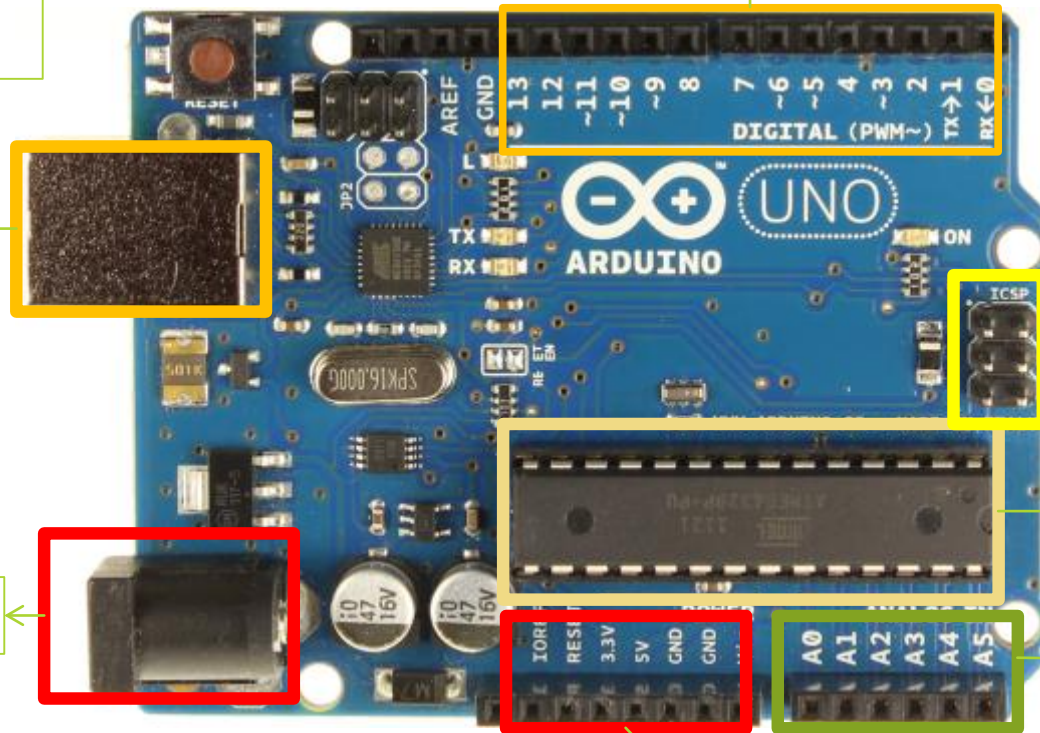
Power input

In circuit
Serial
programming

Atmel
MicroController

Analog input

Power Supply



Arduino code structure:

`Void setup(){}`

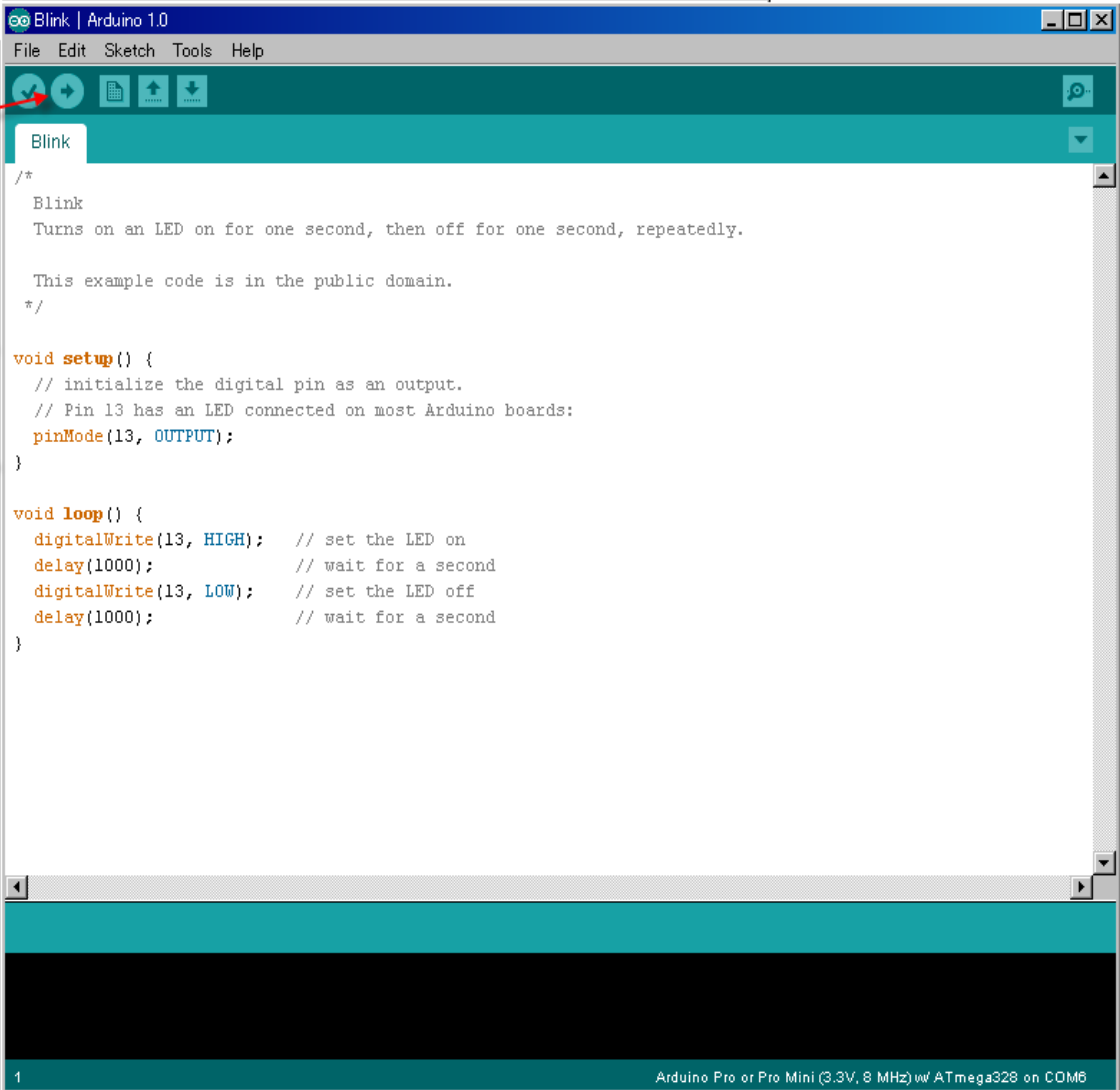
Used to indicate the initial values of system on starting.

`Void loop(){}`

Contains the statements that will run whenever the system is powered after setup.

Compile +
Download

Two main
code blocks



```
Blink | Arduino 1.0
File Edit Sketch Tools Help

Blink

/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);             // wait for a second
}

1 Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328 on COM6
```

Input and output:

- ▶ `pinMode();`
- ▶ `digitalRead();`
- ▶ `digitalWrite();`
- ▶ `delay(time_ms);`
- ▶ other functions:
- ▶ `analogRead();`
- ▶ `analogWrite(); //PWM`

Arduino

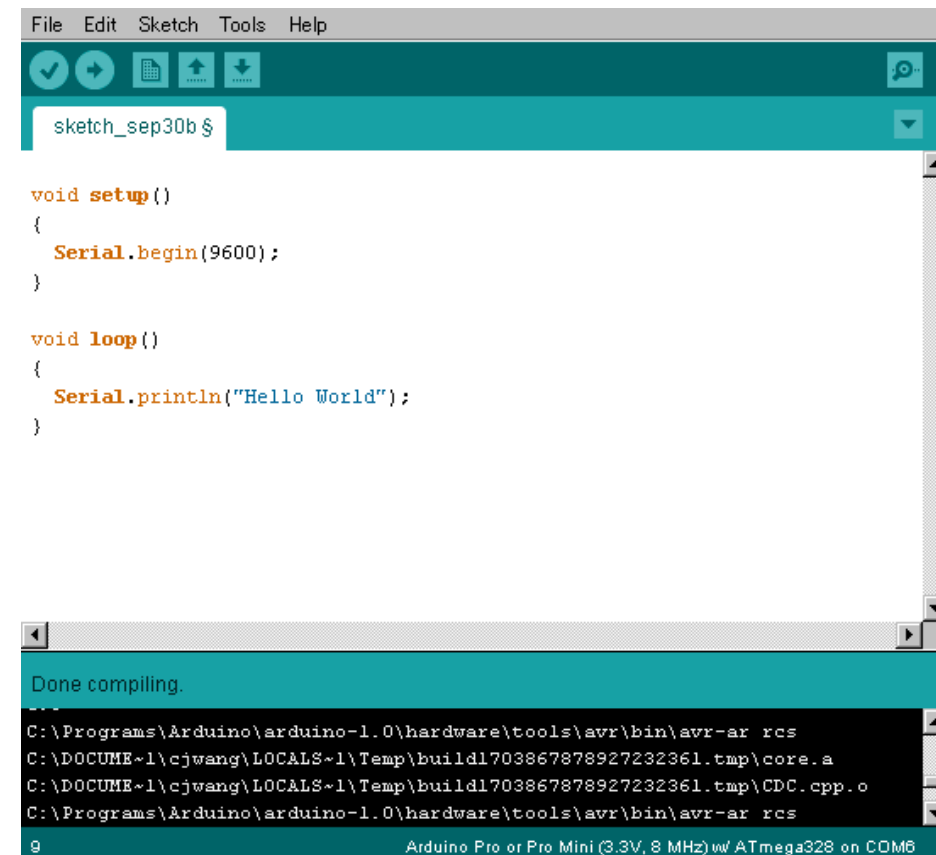
Labs

Lab 1 – Hello World

- ▶ Purpose:
 - ▶ Sanity check
 - ▶ Make sure system is working

Lab 1 – Hello World

- ▶ `Serial.begin(speed)`
 - ▶ initializes serial port at given speed
- ▶ `Serial.print(string)`
 - ▶ prints string
- ▶ `Serial.println(string)`
 - ▶ prints string + trailing newline



The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for checking, saving, opening, and uploading. The main text area contains the following code:

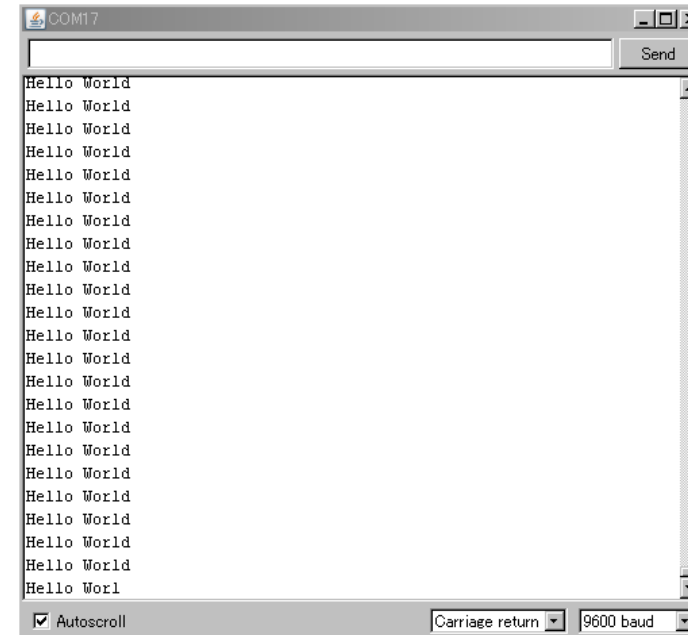
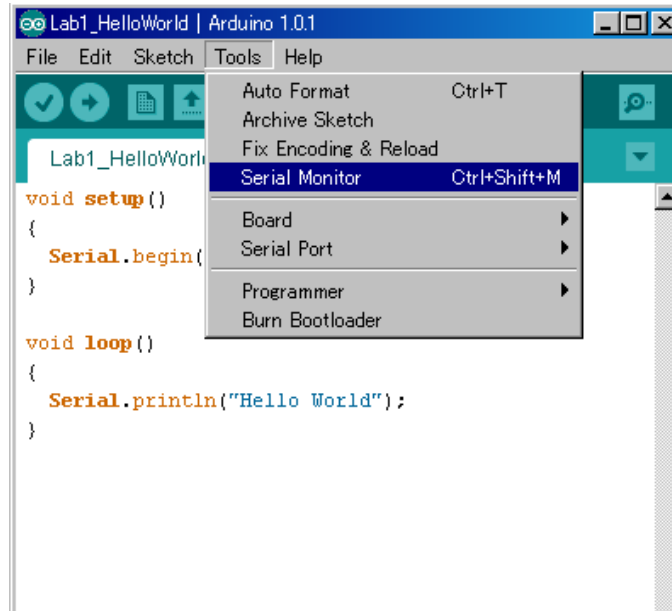
```
sketch_sep30b $  
  
void setup()  
{  
  Serial.begin(9600);  
}  
  
void loop()  
{  
  Serial.println("Hello World");  
}
```

At the bottom, the Serial Monitor window is open, displaying the compilation output:

```
Done compiling.  
C:\Programs\Arduino\arduino-1.0\hardware\tools\avr\bin\avr-ar rcs  
C:\DOCUME~1\cjwang\LOCALS~1\Temp\build1703867878927232361.tmp\core.a  
C:\DOCUME~1\cjwang\LOCALS~1\Temp\build1703867878927232361.tmp\CDC.cpp.o  
C:\Programs\Arduino\arduino-1.0\hardware\tools\avr\bin\avr-ar rcs  
9 Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328 on COM6
```

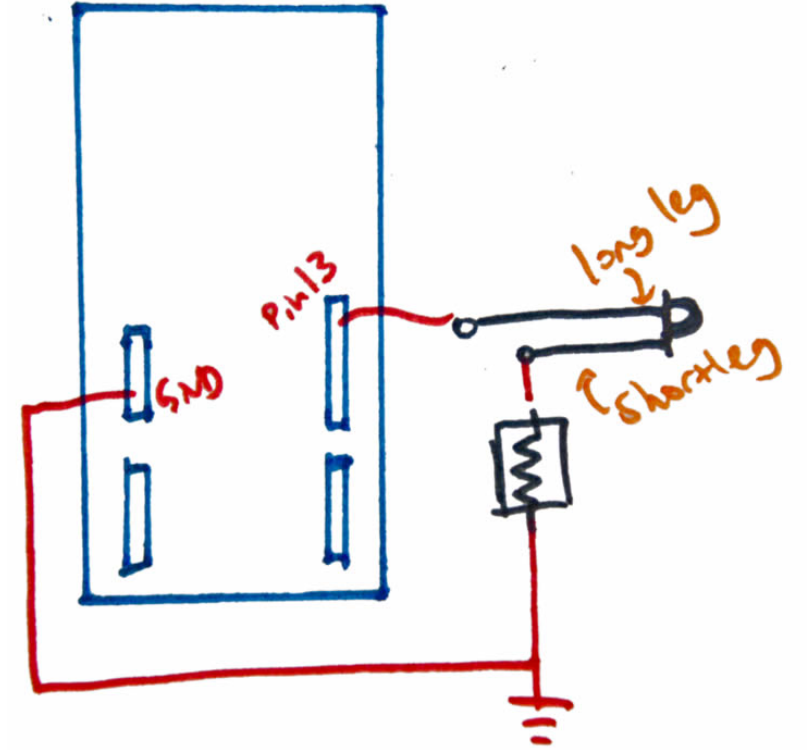
Lab 1 – Hello World

- ▶ You can view serial output from your sketch with the built in serial monitor



Lab 2 - Blink

- ▶ Purpose:
 - ▶ The hello world of embedded
 - ▶ Demonstrate toggling digital voltage levels
 - ▶ If you can blink an LED, you can move the world!



Lab 2 – Blink

- ▶ `pinMode(pin, dir)`
 - ▶ `pin` = pin number
 - ▶ `dir` = INPUT or OUTPUT
- ▶ `digitalWrite(pin, val)`
 - ▶ `pin` = pin number
 - ▶ `val` = LOW or HIGH or 0 or 1
- ▶ `analogWrite(pin, val)`
 - ▶ `pin` = pin number
 - ▶ `val` = 0 to 255
 - ▶ uses PWM
 - ▶ only for certain pins



The screenshot shows the Arduino IDE interface with the 'Blink' sketch loaded. The code is as follows:

```
int ledPin = 13;

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000);                // wait for a second
  digitalWrite(ledPin, LOW);  // set the LED off
  delay(1000);                // wait for a second
}
```

The status bar at the bottom indicates the board is 'Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328 on COM6'.

Digital or Analog?

Analog

- ▶ Analog means that the quantity can take any value between its minimum value and maximum value.
- ▶ All physical quantities are analog.
- ▶ Ex.

Temperature, can take any value[-1,12.8,25.002,... etc.].

- ▶ - Sine waves are analog.

Digital

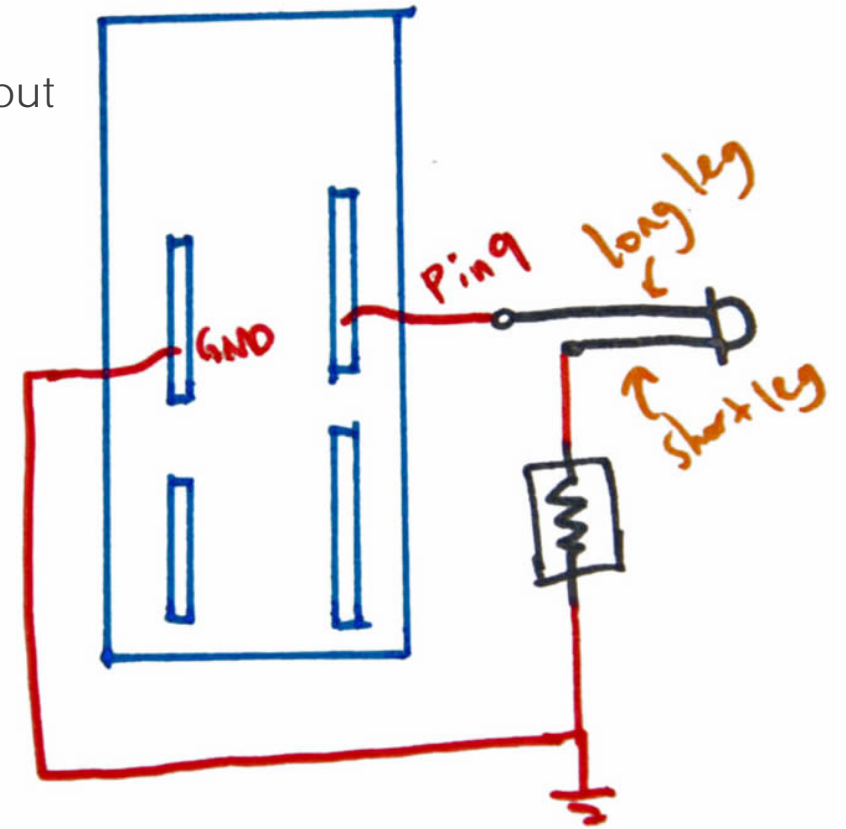
- ▶ Digital means that the quantity can take specific levels of values with specific offset between each other.
- ▶ Ex.

English alpha consists of 26 letter, there is no letter between A and B.

- ▶ - *Square waves are Digital.*

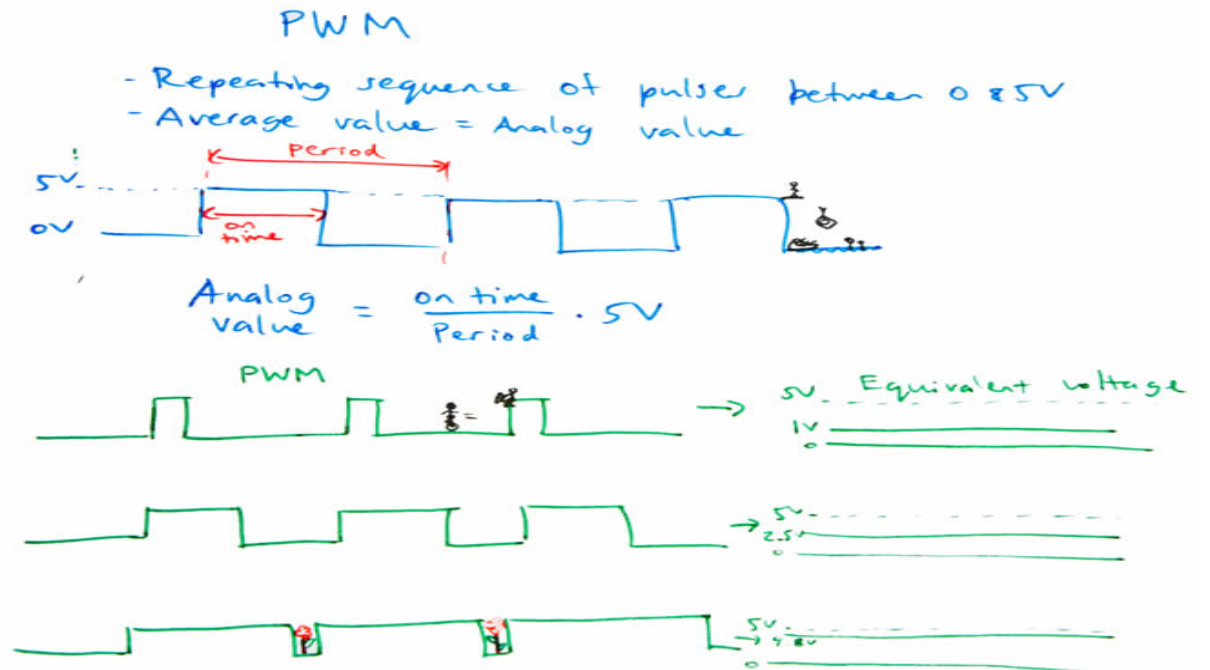
Lab 3 - Fade

- ▶ Purpose:
 - ▶ Demonstrate difference between analog and digital output
 - ▶ Understand pulse width modulation



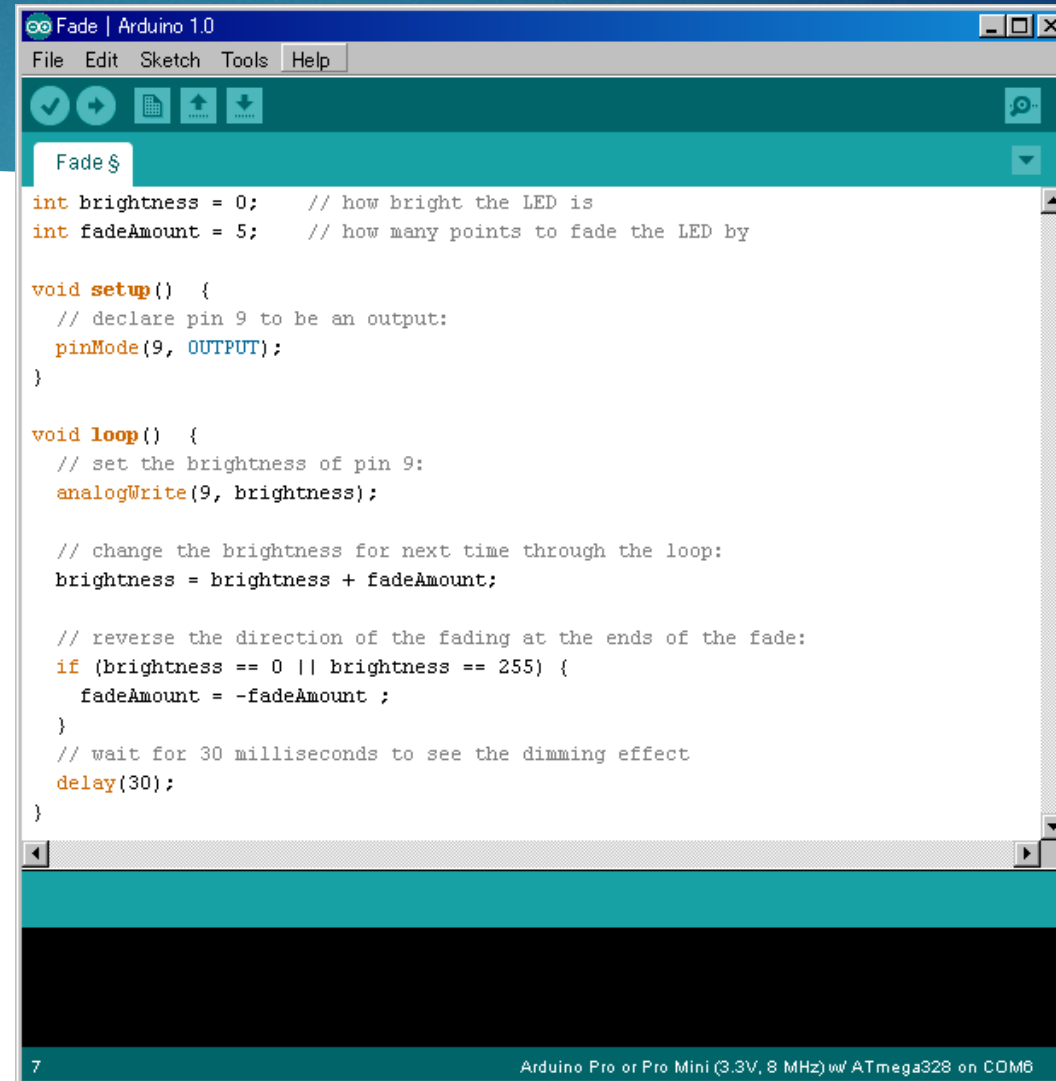
Pulse Width Modulation

- PWM – Method to generate analog voltages from digital voltages



Lab 3 - Fade

- ▶ delay(time)
 - ▶ blocking delay in msec
 - ▶ time = msec



The screenshot shows the Arduino IDE interface with a sketch titled "Fade". The code is as follows:

```
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

void setup() {
  // declare pin 9 to be an output:
  pinMode(9, OUTPUT);
}

void loop() {
  // set the brightness of pin 9:
  analogWrite(9, brightness);

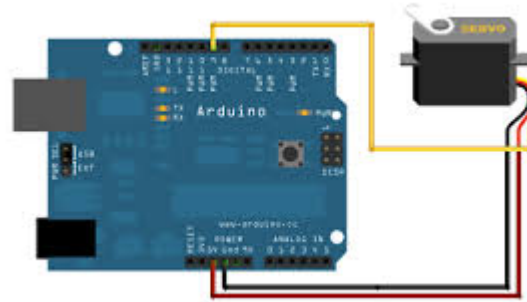
  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

The IDE status bar at the bottom indicates "7" and "Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328 on COM6".

Lab 4 – Motor control using Arduino:

- ▶ Servo motor
- ▶ Applications:
 - ▶ Robotics
 - ▶ Position control



References

► www.arduino.cc





